UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/911,663	07/24/2001	John Edward Ciolfi	04899-060001	3836

7590 02/17/2005

Thomas V. Smurzynski, Esq.
LAHIVE & COCKFIELD LLP
28 State Street
Boston, MA 02109-1784

EXAMINER	
THAI, CUONG T	
ART UNIT	**PAPER NUMBER**
2173	

DATE MAILED: 02/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/911,663	CIOLFI, JOHN EDWARD	
	Examiner	Art Unit	
	CUONG T THAI	2173	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *Amendment filed on July/16/2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *16-46* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *16-46* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

FINAL ACTION

1. This action is responsive to Amendment filed on July/16/2004.

2. Claims 16-46 are presented for examination. Claims 1-15 have been canceled.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102(e) that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 16 and 45 are rejected under 35 U.S.C. 102(e) as being anticipated by McKaskle et al. (USPN: 5,481,741) hereinafter McKaskle.

As per claims 16 (method) and 45 (readable medium), McKaskle discloses a method of mapping graphical block diagram block parameters in a graphical block diagram modeling environment as the technique of the user <u>construct the graphical diagram</u>, the block diagram wherein the block diagram which visually displays a procedure by which a value for a input variable produces a value for one or more output variables (see col. 11, lines 57-62), comprising:

Receiving a user-defined block parameter is taught by McKaskle as the technique of a system for modeling a process (see col. 6, lines 48-49) including the

block diagram editor 30 is **used to construct** and display a graphical diagram, referred to as a block diagram, which visually displays a procedure by which a value for a input variable produces a value for one or more output variables ... the block diagram editor 30 constructs instructions in the machine language which execute the block diagram created by the user (see col. 11, lines 55-67); and

Processing the user-defined block parameter to optimally produce a run-time block is taught by McKaskle as the technique of a system and method, which allows a user to programmatically access various parameters of a control or indicator. In this manner, **a user can programmatically make changes that affects the output or appearance of controls and indicators** (see col. 5, lines 47-51) and the user can view changes to the attribute during execution. Some attributes can be changed by the user and practically all can be changed by the execution subsystems (see col. 6, lines 24-27).

These claims are therefore rejected for the reasons as set forth above.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 17-20, 22-26, 33-38 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle et al. (USPN: 5,481,741) hereinafter McKaskle in view of Dardinski et al. (USPN: 6,754,885 B1) hereinafter Dardinski.

As per claims 33 (method) and 46 (readable medium), McKaskle discloses the inventions substantially as claimed above. McKaskle discloses the limitations of receiving a plurality of user-defined block parameters; processing the plurality of user-defined block parameter to optimally produce a plurality of run-time block parameters as the technique of a system and method, which allows a user to programmatically access various parameters of a control or indicator. In this manner, **a user can programmatically make changes that affects the output or appearance of controls and indicators** (see col. 5, lines 47-51) and the user can view changes to the attribute during execution. Some attributes can be changed by the user and practically all can be changed by the execution subsystems (see col. 6, lines 24-27). McKaskle, however, does not disclose the limitation of pooling together like non-interfaced run-time block parameters to create a run-time parameter expression.

Dardinski discloses the limitation of pooling together like non-interfaced run-time block parameters to create a run-time parameter expression as the techniques of a Parameterized Object consists of a self-contained, cohesive set of parameters when in reality, data inheritance, parameter overrides, and modifications **are all acting together** to determine final parameter values (see col. 11, lines 1-5).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of pooling together like non-

interfaced run-time block parameters to create a run-time parameter expression into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user **pooling all** of a self-contained, cohesive set of parameters when in reality, data inheritance, parameter overrides, and modifications **together**, and thus allowing user capable of controlling object model appearance at any level as desired by the user.

As per claim 17, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of a block method inversely mapping the block run-time parameter to the user-defined block parameter to optimize block implementation.

Dardinski discloses the limitation of a block method inversely mapping the block run-time parameter to the user-defined block parameter to optimize block implementation as the technique of each instance of the Object Type hierarchical which serves as a reference for a Typed Object requires a definition reference to the defining Parameterized Object which defines that type of object... If a user clicks and drags an AOUT definition to a view, then drops it, this relationship provides access to the Parameterized Object which actually defines an AOUT block so that it can be created quickly (see col. 17, lines 42-52 and see Foxboro Defined Object Type and User Defined Object Type in Fig. 13).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of inversely mapping the block run-time parameter to the user-defined block parameter by changing class from Foxboro

Defined Object Type to User Defined Object Type into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to quickly interchange from one class object type to another as desired by the user.

As per claim 18, due to the similarity of this claim to the first limitation of claim 33, this claim is therefore rejected for at least of the same reason applied to claim 33 as set forth above.

As per claim 19, due to the similarity of this claim to the second limitation of claim 33, this claim is therefore rejected for at least of the same reason applied to claim 33 as set forth above.

As per claim 20, due to the similarity of this claim to that of claim 19, this claim is therefore rejected for the same reasons applied to claim 19.

As per claim 22, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of mapping by discarding at least a portion of the plurality of user-defined block parameters to reduce memory requirements.

Dardinski discloses the limitation of discarding at least a portion of the plurality of user-defined block parameters as the technique of add and delete blocks function to sheet (see col. 82, line 23).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to combine add or delete function for adding and deleting function block. Thus, the system would be enhanced by allowing user capability of reducing its required memory.

As per claim 23, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of pooling like non-interfaced runtime block parameters to reduce repetition of the non-interfaced run-time block parameters.

Dardinski discloses the limitation of pooling like non-interfaced runtime block parameters to reduce repetition of the non-interfaced run-time block parameters as the technique of the parameter can be defined in two ways- **by setting the "value" attribute to a constant value. In the "value" attribute of a parameter, user can supply constant default values for parameters in block definitions**(see col. 74, lines 31-32) and a Parameterized Object consists of a self-contained, cohesive set of parameters when in reality, data inheritance, parameter overrides, and modifications **are all acting together** to determine final parameter values (see col. 11, lines 1-5).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of pooling like non-interfaced, constant runtime block parameters into that of McKaskle's invention. By doing so, the system would be enhanced by reducing unnecessarily repetitions and save time for other tasks.

As per claim 24, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of wherein the pooling step comprises mapping user defined block parameters into an existing pool.

Dardinski discloses the limitation of mapping user defined block parameters into an existing pool as the technique of a Parameterized Object consists of a self-contained, cohesive set of parameters when in reality, data inheritance, **parameter overrides**, and **modifications are all acting together** to determine final parameter values (see col. 11, lines 1-5).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of mapping user defined block parameters into an existing pool into that of McKaskle's invention. By doing so, the system would be enhanced by determining final parameter values for overall object model.

As per claim 34, due to the similarity of this claim to that of claim 24, this claim is therefore rejected for the same reasons applied to claim 24.

As per claim 35, wherein the non-interfaced run-time block parameters have stored values that differ from presented values is taught by McKaskle as the technique of the virtual instrument includes a block diagram 46 which graphically provides a specified value for an input variable displayed in the front panel 42 can produce a corresponding value for an output variable in the front panel (see col. 12, lines 22-26)

and equations as Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110) wherein second exponent is constant non-interfaced parameter which different than X and Y variables. This claim is therefore rejected for the reason as set forth above.

As per claim 36, the limitation of wherein the non-interfaced run-time block parameters are fixed point is taught by McKaskle as the technique of equation or Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110) wherein **second exponent is constant** non-interfaced parameter which different than X and Y variables. This claim is therefore rejected for the reason as set forth above.

As per claim 37, McKaskle discloses the invention substantially as claimed above. McKaskle, however, does not disclose the limitation of translating at run-time constant parameter values to an internal representation to enable increased pooling.

Dardinski discloses the limitation of translating at run-time constant parameter values to an internal representation to enable increased pooling as the technique of a Parameterized Object consists of a self-contained, cohesive set of parameters when in reality, data inheritance, parameter overrides, and modifications are **all acting together** to **determine final parameter values** (see col. 11, lines 1-5).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of translating at run-time constant parameter values to an internal representation to enable increased pooling into that of McKaskle's invention. By doing so, the system would be enhanced by capable of

combining all of parameter modification, self-contained or constant parameter, and parameter override in order to finalize the pooling process.

As per claim 38, the limitation of collecting constant portions of an expression containing an interfaced variable is taught by McKaskle as the technique of an expression of Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110) wherein second exponent is constant non-interfaced parameter which different than X and Y variables. This claim is therefore rejected for the reason as set forth above.

As per claim 25, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of wherein the pooling step is repeated with additional optimization.

Dardinski discloses the limitation of wherein the pooling step is repeated with additional optimization as the technique of a Parameterized Object consists of a self-contained, **cohesive set of parameters** when in reality, data inheritance, parameter overrides, and modifications are **all acting together** to determine final parameter values (see col. 11, lines 1-5).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of wherein the pooling step is repeated with additional optimization into that of McKaskle's invention. By doing so, the system would be enhanced by capable of pooling additional repetition for determining final parameter values for overall object model.

As per claim 26, McKaskle discloses the inventions substantially as claimed above. McKaskle, however, does not disclose the limitation of mapping by translating the plurality of user-defined block parameters based at least in part on type.

Dardinski discloses the limitation of mapping by translating the plurality of user-defined block parameters based at least in part on type as the technique of each instance of the Object Type hierarchical which serves as a reference for a Typed Object requires a definition reference to the defining Parameterized Object which defines that type of object... If a user clicks and drags an AOUT definition to a view, then drops it, this relationship provides access to the Parameterized Object which actually defines an AOUT block so that it can be created quickly (see col. 17, lines 42-52 and see **Foxboro Defined Object Type and User Defined Object Type** in Fig. 13).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of translating the plurality of user-defined block parameters based at least in part on type by changing class from Foxboro Defined Object Type to User Defined Object Type into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to quickly interchange from one class object type to another as desired by the user.

7. Claims 21, 27-29, 30-32, 39-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over McKaskle et al. (USPN: 5,481,741) hereinafter McKaskle in view of

Dardinski et al. (USPN: 6,754,885 B1) hereinafter Dardinski and further in view of McDonald et al. (5,966,532) hereinafter McDonald.

As per claim 21, McKaskle-Dardinski discloses the invention substantially as claimed above. McKaskle discloses the limitation of the run-time block parameter is configured to return at least one of simulation results as the technique of the virtual instrument includes a block diagram 46 which graphically provides a specified value for an input variable displayed in the front panel 42 can produce a corresponding value for an output variable in the front panel (see col. 12, lines 22-26) and McKaskle also discloses equations as Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110). McKaskle-Dardinski, however, do not disclose the limitation of automatically generated code that implements model equation.

McDonald discloses the limitation of as the technique of automatically generated code that implements model equation the user initiates a graphical code generation wizard for the control. In the preferred embodiment, the user "pops up" on the control with the mouse and selects the graphical code generation wizard from the menu. Alternatively, the user selects the control from the special palette comprising only the first plurality of controls having an associated graphical code portion or template. In this embodiment, selection of control from this palette automatically initiates the graphical code generation wizard (see col. 4, lines 20-28).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to include McDonald's teaching of automatically generated code that implements model equation into that of McKaskle-Dardinski combined invention. By

doing so, the system would be enhanced by capable of allowing user selects code generation wizard template from menu for generating code from formula equation of object modeling.

As per claim 27, due to the similarity of this claim to that of claim 21, this claim is therefore rejected for the same reasons applied to claim 21.

As per claim 28, McKaskle-Dardinski discloses the invention substantially as claimed above. McKaskle-Dardinski, however, do not disclose the limitation of the parameter expressions are maintained in the automatically generated code.

McDonald discloses the limitation of the parameter expressions are maintained in the automatically generated code as the technique of an in memory snapshot of all tags in the system (see col. 22, line 35).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to include McDonald's teaching of the parameter expressions are maintained in the automatically generated code into that of McKaskle-Dardinski combined invention. By doing so, the system would be enhanced by capable of saving expressions generated from code generation into memory for future use.

As per claim 29, McKaskle discloses the invention substantially as claimed above. McKaskle, however, does not disclose the limitation of wherein the parameter expressions contain interfaced variables that are update-able during modeling.

Dardinski discloses the limitation of wherein the parameter expressions contain interfaced variables that are update-able during modeling as the technique of the attribute values for each parameter can be modified by the user (see col. 73, lines 54-55).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of wherein the parameter expressions contain interfaced variables that are update-able during modeling into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to modify parameter values based on user's desired choice.

As per claim 30, McKaskle discloses convert to a relative more compact portion of the parameter expressions that are constants as the technique of parameter equations as Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110) wherein second exponent is constant parameter. McKaskle, however, does not disclose the limitation of while providing access to interface variables that are update-able.

Dardinski discloses the limitation of providing access to interface variables that are update-able as the technique of the attribute values for each parameter can be **modified by the user** (see col. 73, lines 54-55).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of providing access to interface variables that are update-able into that of McKaskle's compact parameter expression

invention. By doing so, the system would be enhanced by allowing user to modify parameter values based on user's desired choice.

As per claim 31, due to the similarity of this claim to that of claim 29 below, this claim is therefore rejected for the same reasons applied to claim 29.

As per claim 32, McKaskle discloses the invention substantially as claimed above. McKaskle, however, does not disclose the limitation of wherein updateable variables used in a plurality of blocks declared only once.

Dardinski discloses the limitation of wherein updateable variables used in a plurality of blocks declared only once as the technique of the attribute values for each parameter can be modified by the user. However, some inherited parameters can not be override (see col. 73, lines 54-55).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to modify parameter values based on user's desired choice.

As per claim 39, McKaskle discloses the invention substantially as claimed above. McKaskle discloses the limitation of the run-time block parameter is configured to return at least one of simulation results as the technique of the virtual instrument includes a block diagram 46 which graphically provides a specified value for an input variable displayed in the front panel 42 can produce a corresponding value for an output

variable in the front panel (see col. 12, lines 22-26) and McKaskle also discloses equations as Formula Node of Y= X ^ 2 + x + 1 (see Fig. 110). McKaskle-Dardinski, however, do not disclose the limitation of automatically generated code that implements model equation.

McDonald discloses the limitation of as the technique of automatically generated code that implements model equation the user initiates a graphical code generation wizard for the control. In the preferred embodiment, the user "pops up" on the control with the mouse and selects the graphical code generation wizard from the menu. Alternatively, the user selects the control from the special palette comprising only the first plurality of controls having an associated graphical code portion or template. In this embodiment, selection of control from this palette automatically initiates the graphical code generation wizard (see col. 4, lines 20-28).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to include McDonald's teaching of automatically generated code that implements model equation into that of McKaskle-Dardinski combined invention. By doing so, the system would be enhanced by capable of allowing user selects code generation wizard template from menu for generating code from formula equation of object modeling.

As per claim 40, McKaskle-Dardinski disclose the invention substantially as claimed above. McKaskle-Dardinski, however, do not discloses the limitation of the parameter expressions are maintained in the automatically generated code.

McDonald discloses the limitation of the parameter expressions are maintained in the automatically generated code as the technique of an in memory snapshot of all tags in the system (see col. 22, line 35).

It would have been obvious to one having ordinary skill in the art at the time the invention was made to include McDonald's teaching of the parameter expressions are maintained in the automatically generated code into that of McKaskle-Dardinski combined invention. By doing so, the system would be enhanced by capable of saving expressions generated from code generation into memory for future use.

As per claim 41, McKaskle discloses the invention substantially as claimed above. McKaskle, however, does not discloses the limitation of wherein the parameter expressions contain interfaced variables that are updatable during modeling.

Dardinski discloses the limitation of wherein the parameter expressions contain interfaced variables that are updatable during modeling as the technique of the attribute values for each parameter can be modified by the user (see col. 73, lines 54-55).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of wherein the parameter expressions contain interfaced variables that are updatable during modeling into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to modify parameter values based on user's desired choice.

As per claim 42, due to the similarity of this claim to that of claim 30, this claim is therefore rejected for the same reasons applied to claim 30.

As per claim 43, due to the similarity of this claim to that of claim 41, this claim is therefore rejected for the same reasons applied to claim 41.

As per claim 44, McKaskle discloses the invention substantially as claimed above. McKaskle, however, does not disclose the limitation of wherein updateable variables used in a plurality of blocks declared only once.

Dardinski discloses the limitation of wherein updateable variables used in a plurality of blocks declared only once as the technique of the attribute values for each parameter can be modified by the user. However, some inherited parameters can not be override (see col. 73, lines 54-55).

It would have been obvious to one ordinary skill in the art at the time the invention was made to include Dardinski's teaching of into that of McKaskle's invention. By doing so, the system would be enhanced by allowing user to modify parameter values based on user's desired choice.

8. Applicant's arguments filed on July 16, 2004 have been fully considered, but they are not persuasive.

On the last paragraph of page 9 to the first paragraph of page 10, Applicant argues that " Applicant respectfully submits that the combination of McKaskle and McDonald fails to

each or suggest a method of, "mapping graphical block diagram block parameters in a graphical block diagram modeling environment" that includes "processing the user-defined block parameters to optimally produce a run-time block parameter.", and the combination of McKaskle and McDonald fall short of teaching or suggesting the claimed invention". The Examiner, however, does not agree to this argument because the limitation of mapping graphical block diagram block parameters in a graphical block diagram modeling environment is taught by McKaskle as the technique of the user **construct the graphical diagram**, the block diagram wherein the block diagram which visually displays a procedure by which a value for a input variable produces a value for one or more output variables (see col. 11, lines 57-62). McKaskle further discloses the limitation of processing the user-defined block parameters to optimally produce a run-time block parameter as the technique of a system and method, which allows a user to programmatically access various parameters of a control or indicator. In this manner, **a user can programmatically make changes that affects the output or appearance of controls and indicators** (see col. 5, lines 47-51) and the user can view changes to the attribute during execution. Some attributes can be changed by the user and practically all can be changed by the execution subsystems (see col. 6, lines 24-27). Thus, the system would be enhanced by allowing user capability of accessing and changing a plurality of block parameters during the execution processing and this process, in turn, changing affects of the output appearance of control indicator.

Conclusion

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CUONG T THAI whose telephone number is (571) 272-4056. The examiner can normally be reached on 8:00 am - 4:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John W. Cabeca can be reached at (571) 272-4048. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CUONG T THAI
Examiner
Art Unit 2173

***
February 14,2005

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173